

# IBM Open Liberty Grundlagen



JEE Microservices

@ CGS IT – 2023

Version 1.0.5

# Inhalt

- Was ist Open Liberty
- Open Liberty - Architektur
- Open Liberty – Projekt erstellung

# Open Liberty

- A lightweight open framework for building fast and efficient cloud-native Java microservices.
- Build cloud-native apps and microservices while running only what you need. Open Liberty™ is the most flexible server runtime available to Java™.

# Open Liberty

- Open Liberty bietet einen Wizard an um das Startprojekt entsprechend schnell generieren zu können.

<https://openliberty.io/start/>

## Create a starter application

Select the development tools that you prefer to use, then generate a package to start developing your application.  
Find a bug? Need an enhancement? [Raise an issue.](#)

Group

at.cgsit.kursdemo

Artifact

kursopenliberty

Build Tool

☒ Maven ☐ Gradle

Java SE Version

11

Java EE/Jakarta EE Version

10

MicroProfile Version

6.0

Generate Project

# Open Liberty

- Maven commands wie bei Quarkus
- Server Starup Console bietet die Einstiegs URLs bereits an

```
liberty:clean
liberty:compile-jsp
liberty:configure-arquillian
liberty:create
liberty:debug
liberty:deploy
liberty:dev
liberty:devc
liberty:display-url
liberty:dump
liberty:generate-features
liberty:help
liberty:install-feature
liberty:install-server
liberty:java-dump
liberty:package
liberty:prepare-feature
liberty:run
liberty:start
liberty:status
liberty:stop
liberty:test-start
liberty:test-stop
liberty:undeploy
liberty:uninstall-feature
```

ation der ID com.ibm.ws.secur

'keystore\_password' erstellt

lasses because bootstrap cla

localhost:2809/NameService ver

st:9080/openapi/

ult\_host): <http://localhost:9080/ibm/api/>

ult\_host): <http://localhost:9080/openapi/ui/>

ult\_host): <http://localhost:9080/health/>

ult\_host): <http://localhost:9080/metrics/>

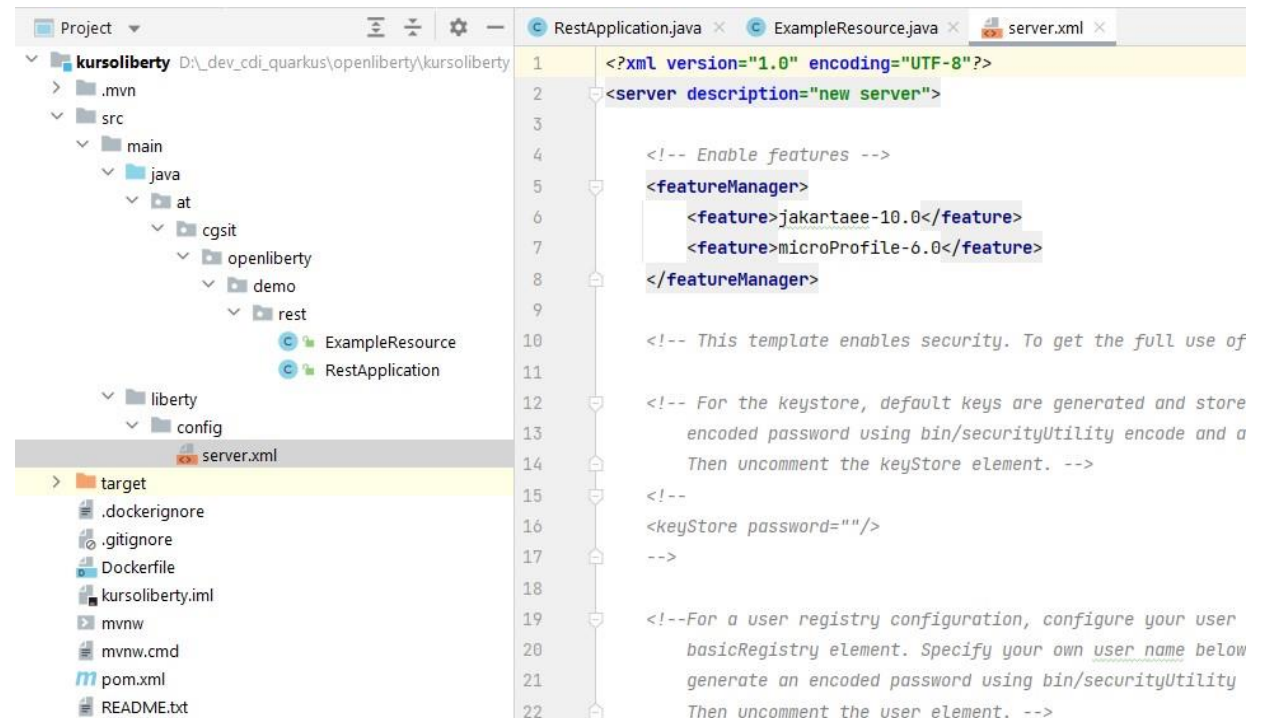
ult\_host): <http://localhost:9080/jwt/>

ult\_host): <http://localhost:9080/kursoliberty/>

t nach 5,686 Sekunden gestartet.

# Open Liberty

- Konfiguration via server.xml file



The screenshot displays an IDE interface with a project explorer on the left and a code editor on the right. The project explorer shows the following structure:

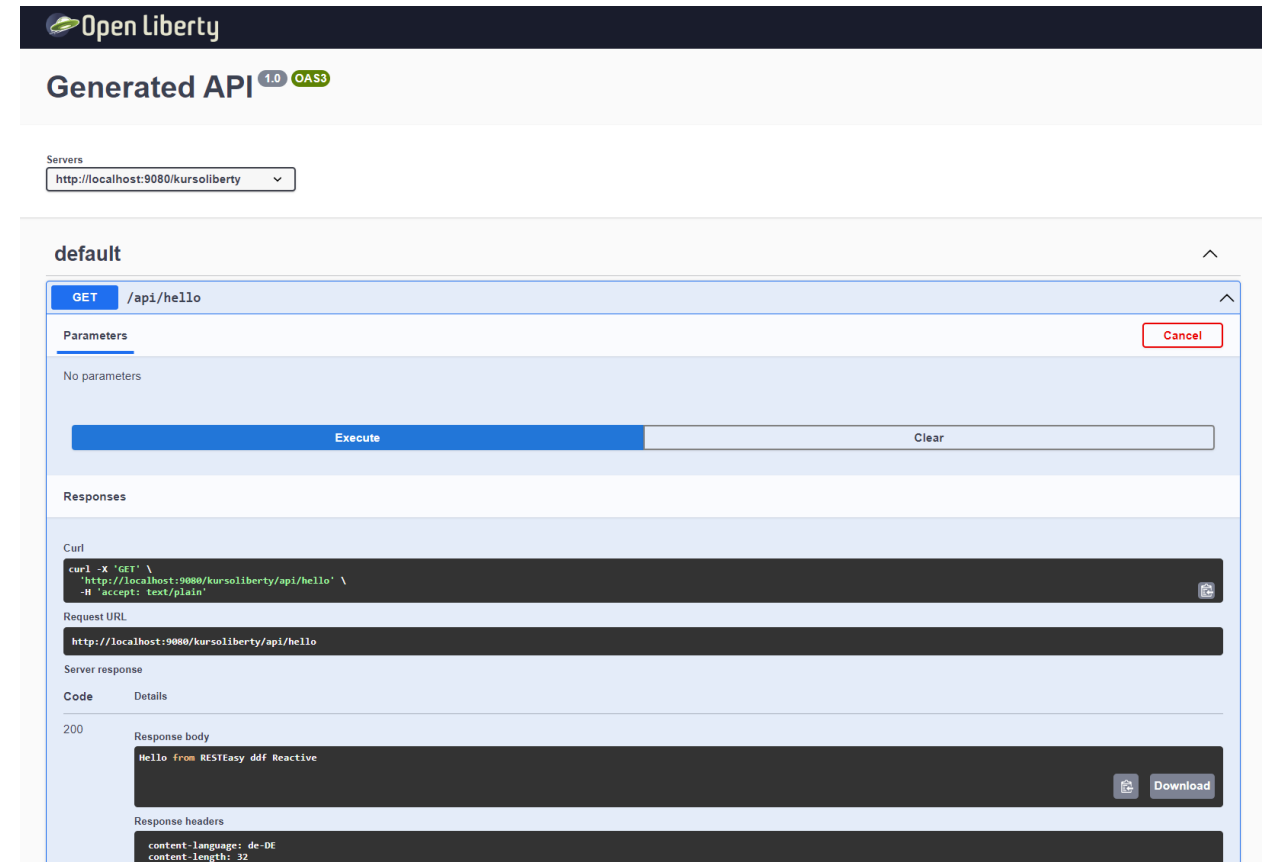
- Project: kursoliberty (D:\\_dev\_cdi\_quarkus\openliberty\kursoliberty)
  - .mvn
  - src
    - main
      - java
        - at
          - cgsit
            - openliberty
              - demo
                - rest
                  - ExampleResource
                  - RestApplication
  - liberty
    - config
      - server.xml
  - target
    - .dockerignore
    - .gitignore
    - Dockerfile
    - kursoliberty.iml
    - mvnw
    - mvnw.cmd
    - pom.xml
    - README.txt

The code editor shows the content of the `server.xml` file:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <server description="new server">
3
4     <!-- Enable features -->
5     <featureManager>
6         <feature>jakartaee-10.0</feature>
7         <feature>microProfile-6.0</feature>
8     </featureManager>
9
10    <!-- This template enables security. To get the full use of
11
12    <!-- For the keystore, default keys are generated and store
13        encoded password using bin/securityUtility encode and a
14        Then uncomment the keyStore element. -->
15
16    <!--
17    <keyStore password=""/>
18    -->
19
20    <!--For a user registry configuration, configure your user
21        basicRegistry element. Specify your own user name below
22        generate an encoded password using bin/securityUtility
23        Then uncomment the user element. -->
```

# Open Liberty

- Swagger console
- OpenAPI als public url verfügbar
- Health und Metrics auch.



Danke für Ihre Aufmerksamkeit